An Interop Publication

ConneXions®

The Interoperability Report

February 1994 Volume 8, No. 2

ConneXions — The Interoperability Report tracks current and emerging standards and technologies within the computer and communications industry.

In this issue:

ConneXions is published monthly by Interop Company, a division of ZD Expos, 303 Vintage Park Drive, Foster City, California, 94404–1138, USA.
Phone: +1 (415) 578-6900
Fax: +1 (415) 525-0194
E-mail: connexions@interop.com

ISSN 0894–5926

From the Editor

Let me start out by apologizing for the lateness of the January issue. You probably received it before the end of the month, but just barely. We are still getting life organized in our new facilities, and things should return to normal soon. The new subscription database is also being developed and we hope to have it on line by April. Meanwhile, let me once again remind you of our new address:

303 Vintage Park Drive, Suite 201
Foster City, CA 94404–1138, USA
Phone: +1 415-578-6900 Fax: +1 415-525-0194
E-mail: `connexions@interop.com`

Richard Stevens recently wrote a book entitled "TCP/IP Illustrated, Volume 1: The Protocols," published by Addison-Wesley. In this issue we bring you an adaptation of the chapter on *TCP Keepalives,* a controversial mechanism employed by many implementations. Since TCP does not use any polling mechanism, no data flows across an idle connection, and it may stay open "forever," unaware of link- or router outages, so long as the TCP processes at each end remain intact. The keepalive feature is intended to detect if the client's host has crashed, or if there are half-open connections.

The Internet suite of protocols has not traditionally been known for ease of use, and plug-and-play installations. As we have discussed in several past issues of *ConneXions,* this "limitation" is disappearing as new tools and application protocols are developed. (Look for articles on *WAIS, Archie, Gopher, WordWideWeb, Prospero, Netfind,* and *DHCP* in the *ConneXions* index). This month we bring you yet another example of this trend, namely an article on the *Service Location Protocol* being developed by the IETF. The service location mechanism allows users to locate resources on the network. It is described here by Scott Kaplan and John Veizades of FTP Software.

The network publishing system, WAIS has been discussed in a previous issue of *ConneXions.* This month, Margaret St. Pierre of WAIS, Inc., describes the underlying protocol for WAIS, a subset of Z39.50-1988. Development activity for the next generation WAIS systems is also presented.

Jon Crowcroft has been a regular contributor to *ConneXions* since 1988. This month, he changes gear a bit and takes us on a journey entitled "Seven Steps to Cyberspace."

The list of Internet books on pages 24 and 25 was originally intended as a footnote to the book review on page 23, but as I quickly discovered there really are *lots* of books on this topic. We will publish an updated list before the end the year.

TCP Keepalives

by W. Richard Stevens

"The use of a feature (X-level NOP) to test the liveness of a TCP connection is consonant with the model against which the TCP was designed."

—*Vint Cerf* [1]

"The use of keepalives is terrible, but sometimes necessary."

—*Dave Crocker* [2]

"Oh what fun! Keepalive wars return.... Well, I'm a firm hater of keepalives, although Mike Karels has persuaded me that in the current world they are a useful tool for catching clients that go off into hyperspace without telling you. I have lots of fellow travellers (actually, I'm probably a fellow traveller with Phil Karn, president of the "I hate keep-alives" party) ..."

—*Craig Partridge* [3]

Introduction

Many newcomers to TCP/IP are surprised to learn that no data whatsoever flows across an idle TCP connection. That is, if neither process at the ends of a TCP connection is sending data to the other, nothing is exchanged between the two TCP modules. There is no polling, for example, as you might find with other networking protocols. This means we can start a client process that establishes a TCP connection with a server, and walk away for hours, days, weeks or months, and the connection remains up. Intermediate routers can crash and reboot, phone lines may go down and back up, but as long as neither host at the ends of the connection reboots, the connection remains established.

This assumes that neither application—the client or server—has application-level timers to detect inactivity, causing either application to terminate. For example, BGP sends an application probe to the other end every 30 seconds. This is an application timer that is independent of the TCP keepalive timer.

There are times, however, when a server wants to know if the client's host has either crashed and is down, or crashed and rebooted. The keepalive timer, a feature of many implementations, provides this capability.

Cons

Keepalives are not part of the TCP specification. The *Host Requirements RFC* [See RFC 1122 and 1123] provides three reasons not to use them: (1) they can cause perfectly good connections to be dropped during transient failures, (2) they consume unnecessary bandwidth, and (3) they cost money on an internet that charges by the packet. Nevertheless, many implementations provide the keepalive timer.

The keepalive timer is a controversial feature, as the quotes at the beginning of this article indicate. Many feel that this polling of the other end has no place in TCP and should be done by the application, if desired. This is one of the religious issues, because of the fervor expressed by some on the topic.

The keepalive option can cause an otherwise good connection between two processes to be terminated because of a temporary loss of connectivity in the network joining the two end systems. For example, if the keepalive probes are sent during the time that an intermediate router has crashed and is rebooting, TCP will think that the client's host has crashed, which is not what has happened.

The keepalive feature is intended for server applications that might tie up resources on behalf of a client, and want to know if the client host crashes. Many versions of the *Telnet* server and *Rlogin* server enable the keepalive option by default.

Pros

A common example showing the need for the keepalive feature nowadays is when personal computer users use TCP/IP to login to a host using *Telnet*. If they just power off the computer at the end of the day, without logging off, they leave a half-open connection on the server. Sending data across a half-open connection causes a reset to be returned, but if nothing is sent by the end that still has its half of the connection open, that end (the server in this example) will never know that the other end (the client) has gone. In this *Telnet* scenario, if the client disappears, leaving the half-open connection on the server's end, and the server is waiting for some data from the client, the server will wait forever. The keepalive feature is intended to detect these half-open connections from the server side.

Description

In this description we'll call the end that enables the keepalive option the *server,* and the other end the *client.* There is nothing to stop a client from setting this option, but normally it's set by servers. It can also be set by both ends of a connection, if it's important for each end to know if the other end disappears. (When NFS uses TCP, both the client and server normally set this option. With *Rlogin* and *Telnet,* however, only the servers set the option, not the clients.)

If there is no activity on a given connection for 2 hours, the server sends a probe segment to the client. (We'll see what the probe segment looks like in the examples that follow.) The client host must be in one of four states.

1. The client host is still up and running and reachable from the server. The client's TCP responds normally and the server knows that the other end is still up. The server's TCP will reset the keepalive timer for 2 hours in the future. If there is application traffic across the connection before the next 2-hour timer expires, the timer is reset for 2 hours in the future, following the exchange of data.

2. The client's host has crashed and is either down or in the process of rebooting. In either case, its TCP is not responding. The server will not receive a response to its probe and it times out after 75 seconds. The server sends a total of 10 of these probes, 75 seconds apart, and if it doesn't receive a response, the server considers the client's host as down and terminates the connection.

3. The client's host has crashed and rebooted. Here the server will receive a response to its keepalive probe, but the response will be a reset, causing the server to terminate the connection.

4. The client's host is up and running, but unreachable from the server. This is the same as scenario 2, because TCP can't distinguish between the two. All it can tell is that no replies are received to its probes.

The server does not have to worry about the client's host being shut down and then rebooted. (This refers to an operator shutdown, instead of the host crashing.) When the system is shut down by an operator, all application processes are terminated (i.e., the client process), which causes the client's TCP to send a FIN on the connection. Receiving the FIN would cause the server's TCP to report an end-of-file to the server process, allowing the server to detect this scenario.

continued on next page

TCP Keepalives *(continued)*

In the first scenario the server application has no idea that the keepalive probes are taking place. Everything is handled at the TCP layer. It's transparent to the application until one of scenarios 2, 3, or 4 occurs. In these three scenarios an error is returned to the server application by its TCP. (Normally the server has issued a read from the network, waiting for data from the client. If the keepalive feature returns an error, it is returned to the server as the return value from the read.) In scenario 2 the error is something like "connection timed out," and in scenario 3 we expect "connection reset by peer." The fourth scenario may look like the connection timed out, or may cause another error to be returned, depending on whether an ICMP error related to the connection is received. We look at all four scenarios in the next section.

Idle time value

A perpetual question by people discovering the keepalive option is whether the 2-hour idle time value can be changed. They normally want it much lower, on the order of minutes. The value can usually be changed, but in most systems derived from the BSD networking code, the keepalive interval is a system-wide value, so changing it affects all users of the option.

The Host Requirements RFC says that an implementation *may* provide the keepalive feature, but it *must* not be enabled unless the application specifically says so. Also, the keepalive interval must be configurable, but it must default to no less than 2 hours.

We'll now look at examples of scenarios 2, 3, and 4, to see the packets exchanged using the keepalive option.

Other end crashes

Let's see what happens when the client host crashes and does not reboot. To simulate this we'll do the following steps:

- Establish a connection between a client program on the host **bsdi** and the standard *echo* server on the host **svr4**. Our client program is named *sock* and just copies standard input to the network, and the network to standard output. We could have used the UNIX *Telnet* client to do this, however we must be able to selectively turn on the keepalive option from the client. We enable the keepalive option with the `-K` option.
- Verify that data can go across the connection.
- Watch the client's TCP send keepalive packets every 2 hours, and see them acknowledged by the server's TCP.
- Disconnect the Ethernet cable from the server, and leave it off until the example is complete. This makes the client think the server host has crashed.
- We expect the server to send 10 keepalive probes, 75 seconds apart, before declaring the connection dead.

Here is the interactive output on the client:

```
bsdi % sock -K svr4 echo    -K for keepalive option
hello, world                type this at beginning, to verify
                            connection is up
hello, world                and see this echoed
                            disconnect Ethernet cable after 4 hours
read error: Connection timed out
                            this happens about 6 hours and 10
                            minutes after start
```

Figure 1 shows the *tcpdump* output. (We have removed the connection establishment and the window advertisements. The first number on each line is the line number, for the discussion that follows. The second number is the time in seconds from the first line, and the third number in parentheses is the time difference in seconds from the previous line.)

```
 1     0.0                     bsdi.1055 > svr4.echo: P 1:14(13) ack 1
 2     0.006105 (      0.0061) svr4.echo > bsdi.1055: P 1:14(13) ack 14
 3     0.093140 (      0.0870) bsdi.1055 > svr4.echo: . ack 14

 4  7199.972793 (7199.8797) arp who-has svr4 tell bsdi
 5  7199.974878 (   0.0021) arp reply svr4 is-at 0:0:c0:c2:9b:26
 6  7199.975741 (   0.0009) bsdi.1055 > svr4.echo: . ack 14
 7  7199.979843 (   0.0041) svr4.echo > bsdi.1055: . ack 14

 8 14400.134330 (7200.1545) arp who-has svr4 tell bsdi
 9 14400.136452 (   0.0021) arp reply svr4 is-at 0:0:c0:c2:9b:26
10 14400.137391 (   0.0009) bsdi.1055 > svr4.echo: . ack 14
11 14400.141408 (   0.0040) svr4.echo > bsdi.1055: . ack 14

12 21600.318309 (7200.1769) arp who-has svr4 tell bsdi
13 21675.320373 (  75.0021) arp who-has svr4 tell bsdi
14 21750.322407 (  75.0020) arp who-has svr4 tell bsdi
15 21825.324460 (  75.0021) arp who-has svr4 tell bsdi
16 21900.436749 (  75.1123) arp who-has svr4 tell bsdi
17 21975.438787 (  75.0020) arp who-has svr4 tell bsdi
18 22050.440842 (  75.0021) arp who-has svr4 tell bsdi
19 22125.432883 (  74.9920) arp who-has svr4 tell bsdi
20 22200.434697 (  75.0018) arp who-has svr4 tell bsdi
21 22275.436788 (  75.0021) arp who-has svr4 tell bsdi
```

Figure 1: Keepalive packets that determine that a host has crashed.

Lines 1, 2, and 3 send the line "hello, world" from the client to the server and back. The first keepalive probe occurs 2 hours (7200 seconds) later on line 4. The first thing we see is an ARP request and an ARP reply, before the TCP segment on line 6 can be sent. The keepalive probe on line 6 elicits a response from the other end (line 7). The same sequence of packets is exchanged 2 hours later in lines 8–11.

If we could see all the fields in the keepalive probes, lines 6 and 10, we would see that the sequence number field is one less than the next sequence number to be sent (i.e., 13 in this example, when it should be 14), but because there is no data in the segment, *tcpdump* does not print the sequence number field. (It only prints the sequence number for empty segments that contain the SYN, FIN, or RST flags.) It is the receipt of this incorrect sequence number that forces the server's TCP to respond with an ACK to the keepalive probe. The response tells the client the next sequence number that the server is expecting (14).

Some older implementations based on 4.2BSD do not respond to these keepalive probes unless the segment contains data. Some systems can be configured to send one garbage byte of data in the probe to elicit the response. The garbage byte causes no harm, because it's not the expected byte (it's a byte that the receiver has previously received and acknowledged), so it's thrown away by the receiver. Other systems send the 4.3BSD-style segment (no data) for the first half of the probe period, and if no response is received, switch to the 4.2BSD-style segment for the last half.

continued on next page

TCP Keepalives *(continued)*

We then disconnect the cable and expect the next probe, 2 hours later, to fail. When this next probe takes place, notice that we never see the TCP segments on the cable, because the host is not responding to ARP requests. We can still see that the client sends 10 probes, spaced 75 seconds apart, before giving up. We can see from our interactive script that the error code returned to the client process by TCP gets translated into "Connection timed out," which is what happened.

Other end crashes and reboots

In this example we'll see what happens when the client crashes and reboots. The initial scenario is the same as before, but after we verify that the connection is up, we disconnect the server from the Ethernet, reboot it, and then reconnect it to the Ethernet. We expect the next keepalive probe to generate a reset from the server, because the server now knows nothing about this connection. Here is the interactive session:

```
bsdi % sock -K svr4 echo   -K to enable keepalive option
hi there                   type this to verify connection is up
hi there                   and this is echoed back from other end
                           here server is rebooted while
                           disconnected from Ethernet
read error: Connection reset by peer
```

Figure 2 shows the *tcpdump* output. (We have removed the connection establishment and the window advertisements.)

```
1    0.0                   bsdi.1057 > svr4.echo: P 1:10(9) ack 1
2    0.006406 (    0.0064) svr4.echo > bsdi.1057: P 1:10(9) ack 10
3    0.176922 (    0.1705) bsdi.1057 > svr4.echo: . ack 10

4 7200.067151 (7199.8902) arp who-has svr4 tell bsdi
5 7200.069751 (    0.0026) arp reply svr4 is-at 0:0:c0:c2:9b:26
6 7200.070468 (    0.0007) bsdi.1057 > svr4.echo: . ack 10
7 7200.075050 (    0.0046) svr4.echo > bsdi.1057: R 1135563275:1135563275(0)
```

Figure 2: Keepalive example when other host has crashed and rebooted.

We establish the connection and send 9 bytes of data from the client to the server (lines 1–3). Two hours later the first keepalive probe is sent by the client, and the response is a reset from the server. The client application prints the error "Connection reset by peer," which makes sense.

Other end is unreachable

In this example the client has not crashed, but is not reachable during the 10-minute period when the keepalive probes are sent. An intermediate router may have crashed, a phone line may be temporarily out of order, or something similar.

To simulate this example we'll establish a TCP connection from our host **slip** through our dialup SLIP link to the host **vangogh.cs.berkeley.edu**, and then take the link down. First, here is the interactive output:

```
slip % sock -K vangogh.cs.berkeley.edu echo
testing     we type this line
testing     and see it echoed
            sometime in here the dialup SLIP link is taken down
read error: No route to host
```

W. RICHARD STEVENS holds a B.S.E. in Aerospace Engineering from the University of Michigan and a Ph.D. in Systems Engineering from the University of Arizona. He is the author of the books *UNIX Network Programming* (1990), *Advanced Programming in the UNIX Environment* (1992), and *TCP/IP Illustrated, Volume 1: The Protocols* (1994). He spends most of his time writing, with the remainder spent teaching classes and consulting. He can be reached as `rstevens@noao.edu`.

The connection of the systems is **`slip`** (the client), to **`bsdi`** (where we collect the *tcpdump* output shown in Figure 3), to **`sun`** (which has a dialup SLIP link to the Internet that we'll take down during the test), to the server **`vangogh.`** (The connection establishment and window advertisements have been removed.)

```
1       0.0                      slip.1056 > vangogh.echo: P 1:9(8) ack 1
2       0.277669 (    0.2777)    vangogh.echo > slip.1056: P 1:9(8) ack 9
3       0.424423 (    0.1468)    slip.1056 > vangogh.echo: . ack 9

4    7200.818081 (7200.3937)     slip.1056 > vangogh.echo: . ack 9
5    7201.243046 (    0.4250)    vangogh.echo > slip.1056: . ack 9

6   14400.688106 (7199.4451)     slip.1056 > vangogh.echo: . ack 9
7   14400.689261 (    0.0012)    sun > slip: icmp: net vangogh unreachable
8   14475.684360 (   74.9951)    slip.1056 > vangogh.echo: . ack 9
9   14475.685504 (    0.0011)    sun > slip: icmp: net vangogh unreachable

                               (14 lines deleted)

24  15075.759603 (   75.1008)    slip.1056 > vangogh.echo: R 9:9(0) ack 9
25  15075.760761 (    0.0012)    sun > slip: icmp: net vangogh unreachable
```

Figure 3: Keepalive example when other end is unreachable.

We start the example the same as before: lines 1–3 verify that the connection is up. The first keepalive probe 2 hours later is fine (lines 4 and 5), but before the next one occurs in another 2 hours, we bring down the SLIP connection on the router **`sun`** that is being used for connectivity between the client and server.

The keepalive probe in line 6 elicits an ICMP network unreachable from the router **`sun.`** This is just a soft error to the receiving TCP on the host **`slip`**. It records that the ICMP error was received, but the receipt of the error does not take down the connection. Nine more keepalive probes are sent, 75 seconds apart, before the sending host gives up. The error returned to the application generates a different message this time: "No route to host," which is the BSD error that corresponds to the ICMP network unreachable error.

Summary

As we said earlier, the keepalive feature is controversial. Protocol experts continue to debate whether it belongs in the transport layer, or should be handled entirely by the application. It operates by sending a probe packet across a connection after the connection has been idle for 2 hours. Four different scenarios can occur: the other end is still there, the other end has crashed, the other end has crashed and rebooted, or the other end is currently unreachable. We saw each of these scenarios with an example, and saw different errors returned for the last three conditions.

In the first two examples that we looked at, had this feature not been provided, and without any application-level timer, our client would never have known that the other end had crashed, or crashed and rebooted. In the final example, however, nothing was wrong with the other end, the connection between them was temporarily down. We must be aware of this limitation when using keepalives.

References

[1] Cerf, V., "SO_KEEPALIVE considered harmful?" Message to `comp.protocols.tcp-ip`, 28 May 1989.

[2] Crocker, D., "SO_KEEPALIVE considered harmful?" Message to `comp.protocols.tcp-ip`, 23 May 1989.

[3] Partridge, C., "SO_KEEPALIVE considered harmful?" Message to `comp.protocols.tcp-ip`, 23 May 1989.

The Service Location Protocol:

Making Finding and Sharing Internet Resource a Snap

by Scott Kaplan and John Veizades, FTP Software

Introduction

While many tools have appeared in the Internet to help users navigate through the global maze of information, many users are still grappling with simple tasks such as finding the local printer or a file server and using mail. These users need basic information about their environment, specifically:

- What services are available to meet my needs?
- How do I select from the available services? and
- How do I use the chosen service?

Current solutions to these questions ranges from knocking on your neighbor's door and hoping they can help, to tracking down an administrator and hoping that they can spend a precious few moments to give you the answers.

The Service Location utility solves this problem by using information available on the network to answer these questions. Service Location finds all the services on the network. It queries services to determine their features and can set you up with a requested service.

Protocol overview

The *Service Location Protocol* ("srvloc"), developed by the Internet Engineering Task Force (IETF), provides the following basic set of transactions:

– *Locating services:*

- A client (user agent) broadcasts a request looking for all services
- Each service agent responds with what type of service it is (distinguished attribute)

– *Getting the dictionary:*

- The client requests the dictionary of terms used to talk about the type of service it wants
- The service agent(s) respond with the dictionary(ies)

– *Service request:*

- The client uses the terms in the dictionary to build an expression that describes the service it wants
- Each service agent with a service that matches the client's description responds with the list of attributes that the client wants.

Srvloc relies on the services of a network layer broadcast/multicast datagram delivery protocol like UDP. Though the protocol has been defined in terms of the TCP/IP protocol family, this protocol is easily extended to work in other protocol families (IPX, AppleTalk, etc.) that offer this basic level of service.

Locating services

Typical network clients require a network "wizard" to configure key network parameters. The *Dynamic Host Configuration Protocol* (DHCP) [1] helps simplify some of this configuration, however, finding and configuring the networked client to use services above the network layer is still a daunting task. Remarkably, the information needed for this task is already available from the services on the network.

With srvloc, the administrator allows servers to advertise their presence to clients and to pass them configuration information when necessary. The protocol provides a means for clients to find services, to ask services for the character of the service they provide, and to allow the network user to decide which service is appropriate for their use. This discovery mechanism is dynamic so that as services come and go, clients always have the latest service information. This allows administrators to deal with a small number of services, rather than a larger number of clients.

Services are described formally through the use of attributes. An attribute is a (`class,value`) pair that describes some characteristic of the services. In the tuple, `<class>` is a string and `<value>` is a typed structure. The service agents return an attribute in the initial protocol response. This is the "distinguished attribute." The distinguished attribute's value is a string uniquely identifying a service, the class name is the null value (a null string). Examples include:

```
("", "printer")
("", "modem")
("", "file_server")
("", "mail_server")
```

The distinguished attribute is the start of a taxonomy of attributes that structure a name space. Values for the distinguished attribute are assigned by the *Internet Assigned Numbers Authority* (IANA). To find services in a network, a client multicasts/broadcasts a request for all services to respond. Each service agent multicasts/broadcasts a response to the client's initial request. Service agents hold down before responding and listen for other responses during the hold down period. If a service agent hears all of its distinguished attributes in responses from other service agents, it does not respond.

These responses provide the initial configuration for the client. The client now knows what kinds of services are available in the network.

In addition, each attribute can have a help string and a configuration string associated with it. The configuration string associated with the distinguished attribute provides the network addressing information for the service.

Getting the directory

In order to allow srvloc to describe services that might not exist yet, the srvloc protocol does not define the terms used in the attributes. The client retrieves the dictionary of terms that describes services from the service agents on the network.

Using the distinguished attributes, the user can select a particular set of services (e.g., "printer") and can request the "dictionary" from these services.

The dictionary consists of the terms used in the attributes that describe the service. For example, if a printer has the following database of attributes:

```
("","printer")
("page_desc_lang","ps")
("page_desc_lang","pcl")
("paper_type","plain")
("paper_type","letterhead")
("paper_type","blue")
("page/minute","4")
```

continued on next page

The Service Location Protocol *(continued)*

...then the dictionary would be:

```
page_desc_lang: ps, pcl
paper_type: plain, letterhead, blue
page/minute: 4
```

The client uses the information returned in the dictionary response to formulate a service request, describing the attributes (characteristics) of the service it wants.

The service request

A client specifies the characteristics of the service it wants with a service request. A client can make a service request after it knows the distinguished attribute and dictionary for a type of service. It is also possible to broadcast a service request without making any previous protocol requests if the client has knowledge of the terms used to describe services in some restricted domain.

Assume a site has vending machines. Each vending machine provides a "service," that is, it provides some food-like product. We can use service location to find a snack which meets our culinary needs.

```
("","vending machine")
("drink", "jolt")
("drink", "diet Coke")
("candy", "kit kat")
("candy", "peanut m&m's")
("fruit", "apple")
("fruit", "orange")

(""."vending machine")
("drink", "orange juice")
("drink", "mineral water")
("fruit", "apple")
("fruit", "orange")

(""."vending machine")
("drink", "chocolate milk")
("food", "bean burrito")
("food", "strawberry yogurt")
```

The dictionary for the distinguished attribute value, "vending machine" is:

```
drink: jolt, diet coke, orange juice, mineral water, chocolate milk
candy: kit kat, peanut m&m's
fruit: apple, orange
food: bean burrito, strawberry yogurt
```

A client who is looking for vending machine that has oranges and mineral water or a diet coke sends a service request with the following expression:

```
& ("","vending machine")
  | ("fruit","orange")
    & ("drink","mineral water") ("drink","diet coke")
```

A user interface can translate the user's description into the wire protocol.

Directory agent

As a network implementing srvloc grows, the srvloc protocol begins to represent a large part of the background network traffic. To provide a point of scaling in the network, the srvloc protocol has two features. The first is a central repository of information called a *directory agent* and second is a set of administrative boundaries called *scopes.*

The directory agent is a central repository of information, but unlike most directory services, it is dynamically updated by services in the network. This design avoids the problems that are usually associated with centrally administered directory services like the *Domain Name System* (DNS) and X.500.

Most problems with centralized solutions stem from the fact that they are often manually configured. Common problems include:

- There is a delay between the time the service is established and the time the administrator updates the central authority.
- There are often policy controls on updating the information in a central database and the service description may have to be modified to conform to those policies.
- When services leave the network there can be a substantial delay in updating the central database. During this interval, the central database contains outdated information.

The solution is not to dispense with the centralized repository, but to make sure that it is dynamically loaded by the service agents.

Mechanism

The directory agent is a rendezvous for the service information between the service agent and the user agent. The service agent still has the authoritative information about the service. However, when it starts, it looks for one or more directory agents. If it finds one, it registers all of its information with the directory agent(s). It also notifies the directory agent(s) if its information changes or if it shuts down. Likewise, when the directory agent comes up, it looks for service agents and requests that the service agents register their information.

When a user agent starts looking for information, it first looks for a directory agent. If it finds one, the directory agent becomes the single point of contact for that user agent, bypassing the additional multicasts/broadcasts required to retrieve the information from disparate service agents.

If the directory agent disappears for some reason and user agents fail in their attempts to contact this central repository, they fall back to the behavior they used when a directory agent was not present. In this case, service agents continue to field requests as before.

To allow this protocol to scale to campus wide sites and to allow for redundancy in the directory agent operation, the concept of "scope" was added to the protocol. A scope is an administrative boundary in which an agent can be a member. A directory agent must be present for the scope functionality to be in effect. A directory agent advertises the scopes that are available in a network and a user or service agent picks the scope that is best for them. When a service and a user agent look for a directory agent, they search for a directory agent that supports the scope that they are in. Service agents register with all directory agents that support the scope that they are in; user agents direct queries for resources to directory agents in their default scope.

continued on next page

The Service Location Protocol *(continued)*

A user agent can view all the possible scopes in an internet. The user agent can also look for resources that are in other scopes on their internet. Service agents can be registered in more than one scope allowing them to service the needs of more than one user community.

The Service Location Working Group is planning how this service location scheme would scale in a global environment. It is possible, given the address of a remote directory agent, to perform srvloc queries in a foreign network. The Service Location Working Group is developing codified standards for making this operation seamless to the rest of the srvloc proposal as well as scalable to the Internet of the future.

Example

To understand the benefits of srvloc, let's look at administering a service in a small site with three printers. All printers support the LPD protocol, one of them additionally supports PCNFSD. Two of the printers support only *PostScript*, the other supports both PCL and *PostScript*. The attributes for the three printers are:

Printer A:

```
("","printer")
("print_protocol","lpd")
("page_desc_lang","PostScript")
("page_desc_lang","PCL")
("paper_size","letter")
("paper_size","legal")
("location","lab")
```

Printer B:

```
("","printer")
("print_protocol","lpd")
("page_desc_lang","PostScript")
("paper_size","letter")
("location","receptionist")
```

Printer C:

```
("","printer")
("print_protocol","lpd")
("print_protocol","pcnfsd")
("page_desc_lang","PostScript")
("paper_size","letter")
("paper_size","legal")
("location","lab")
```

Note there is an attribute for the physical location of the printer, "location" which is clearly site-specific.

The configuration string for each attribute in this example is:

```
("","printer")                "IP_addr=128.127.50.3, port=515"
("print_protocol","lpd")
("page_desc_lang","PS")       "<esc>@PLJ ENTER LANGUAGE = PostScript"
("page_desc_lang","PCL")      "<esc>@PLJ ENTER LANGUAGE = PCL"
("paper_size","letter")
("paper_size","legal")
```

Attribute standardization

From the previous example, it is clear that there is a tradeoff between site customization and standardization. Both of these features are highly desirable. Site customization allows an administrator to use a service's attributes to describe features unique to that site. For example, a printer's location attribute might be:

```
("location", "receptionist")
```

While such a description might seem casual and unstructured to a network protocol developer, it is the term that the users most commonly use to describe the printer's location. In fact, administrators spend a large portion of their time translating between terms users understand and the terms the vendor used in defining a network service. Srvloc provides a means for services to advertise these site-specific terms.

On the other hand, this flexibility requires the user to interpret attributes. Ironically, this can reduce the utility of the system to the user because client software can't programatically manipulate attributes without well-defined semantics.

While the wire protocol itself doesn't require us to say anything about the format of the attributes or other strings, to make a functional system we must find a way of supporting "standard" formats as well as providing site-specific extensions.

Standard attributes

To solve this problem, each attribute has an associated byte that describes its "standardization." This "standardization value" is an integral part of the attribute. It is stored in the service's attribute database and it is sent on the wire with the attribute. It is used in computing equality between two attributes. For example, a service which has the attribute:

```
("type","PostScript")     standardization = IETF
```

would not respond to a service request with:

```
("type","PostScript")     standardization = HP
```

The standardization byte covers the attribute class, value, configuration string and help strings. By requesting a particular standardization, the client can be assured of an agreement with a predefined format for the information returned by srvloc. The values for the standardization byte are part of the protocol and are IANA registered.

Operational issues

The srvloc protocol builds networks that are much more manageable than traditional local internets. One of the major costs in maintaining a network of any size is the cost of the personnel who administer the network. In general, any change to the network, or its hosts, may need to be propagated far beyond the area directly affected by the change. For example, adding a printer to the network requires changing configuration on every host that may need to access the printer.

From an operational point of view, srvloc provides a clear definition of the line between LAN and WAN. The "local area" becomes the area you can multicast/broadcast srvloc requests to.

It also means that WAN protocols and tools can focus on tying LAN clusters together, rather than reaching into a LAN and tying all the individual hosts together.

Operationally, the goal is to create a two-level hierarchy, giving administrators a large amount of autonomy in configuring local sites as well as the means to connect sites using existing WAN protocols.

Multilingual support

Modern organizations span many cultural boundaries and many languages. Few distributed network based protocols deal with this fact gracefully.

continued on next page

The Service Location Protocol *(continued)*

The srvloc protocol has been designed so that information can be presented to users in their language of preference. The protocol sends the language requested to the service agent. If the service agent has been loaded with support for that language, the responses are returned in the language of choice. The service agent may also send back responses in the default language if the requested language is not available.

Language support centralizes the translation files at the service. This means end systems need not be configured with specific language files for the services they need to access.

Authentication

Srvloc provides a mechanism to distribute service information far and wide. Policy controls on information are needed to ensure that the source of information is a valid network citizen and that the recipients of information are authorized to view the information. In the case of directory agents, this is more profound concern since a bogus directory agent could bring the complete network infrastructure to its knees. Srvloc provides for an extensible authentication functionality that fits well into many of the authentication models that are being suggested by the IETF *Security Architecture Group* (SAG) as well as *Kerberos* style authentication.

This authentication model allows the user agent to verify the source of srvloc information as well as allowing the service agent to verify the identity of the recipient of the information.

These security models are all optional.

Mailing list

The service location working group is part of the IETF and can be reached at `srv-location@ftp.com`. To be added to the mailing list, send a request to `srv-location-request@ftp.com`. The current Internet Draft (`draft-ietf-svrloc-protocol-02.txt`) of the protocol is available on any of the following Internet servers: `ds.internic.net` (`198.49.45.10`), `venera.isi.edu` (`128.9.0.32`), `munnari.oz.au` (`128.250.1.21`), `nic.nordu.net` (`192.36.148.17`). Beware that Internet Drafts are works-in-progress.

References

[1] Allard, J., "DHCP—TCP/IP Network Configuration Made Easy," *ConneXions,* Volume 7, No. 8, August 1993.

[2] Minshall, G., "AppleTalk versus IP," *ConneXions,* Volume 3, No. 9, September 1989.

[3] Schiller, J., "Kerberos: Network Authentication," *ConneXions,* Volume 4, No. 1, January 1990.

[4] Deutsch, P., "The Internet as Service Provider," *ConneXions,* Volume 7, No. 9, September 1993.

SCOTT KAPLAN (`scott@ftp.com`) is the Director of Engineering at FTP Software's West Coast Operations. He is co-chair of the Service Location Protocol Working Group of the IETF. He is currently studying networking technologies which will make hosts easier to configure and maintain. The ultimate goal of this work is to dispel the myth that grandmothers don't belong on the Internet.

JOHN VEIZADES (`veizades@ftp.com`) is the founder of the Service Location Working Group and is the working groups co-chair. He has been working in the TCP/IP community for the last ten years and has been working with the IETF to bring ease of use and configuration to the TCP/IP protocol family. John previously worked on TCP/IP implementations for personal computers at Xerox and Apple Computer. John is currently the engineering manager of service location work at FTP Software's West Coast Operations.

Wide Area Information Servers (WAIS) over Z39.50-1988 and Beyond

by Margaret St. Pierre, WAIS Incorporated

Introduction

The network publishing system, *Wide Area Information Servers* (WAIS) [1], is designed to help users find information over a computer network. The principles guiding the design of the WAIS system are:

- A wide-area networked-based information system for searching, browsing, and publishing.
- Based on standards.
- Easy to use.
- Flexible and growth oriented.

From the systems that have grown out of these principles, a large group of developers, publishers, standards bodies, libraries, government agencies, educational institutions, and users have been enjoying the benefits of shared information.

WAIS development began in October 1989 with the first Internet release occurring in April 1991. From the beginning, the goal of the WAIS network publishing system was to create an open architecture for information retrieval by using a *standard* computer-to-computer protocol. The underlying protocol was based on the 1988 Version of the NISO (National Information Standards Organization) *American National Standard Z39.50 Information Retrieval Service Definition and Protocol Specifications* [2]. The WAIS implementation is still in use today resulting in over 50,000 users of Z39.50-1988 on the Internet, with an even greater number of users acquiring access via a suite of tools and services, for example, *Gopher, World Wide Web,* and *America OnLine* gateways.

The Z39.50-1988 standard originally grew out of the library community as a search and retrieval protocol for bibliographic data. The design goals for the WAIS system required a more general information search and retrieval protocol. By working with the *Z39.50 Implementor's Group* (ZIG), WAIS developers used a recommended subset of Z39.50-1988 and a specific set of assumptions to fulfill its requirements. Over time, many of these requirements have then gone into the definition of subsequent versions of Z39.50 [3,4].

This article describes the subset of Z39.50-1988 used and the additional assumptions made to meet the design goals of the WAIS network publishing system [5]. In addition, the new development activity taking place on the next generation WAIS systems is also presented.

Concepts

The WAIS architecture has four main components: the *client,* the *server,* the *database,* and the *protocol.* The WAIS client is a user-interface program that sends requests for information to local or remote servers. Clients are available for most popular desktop environments. The WAIS server is a program that services client requests, and is available on a variety of UNIX platforms. The server generally runs on a machine containing one or more information sources, or WAIS databases. The protocol, based on Z39.50, is used to communicate between WAIS clients and servers.

The WAIS system performs two basic operations: *search* and *retrieval.* Each operation is a two-step transaction made up of a request sent by the client to the server followed by a response sent by the server back to the client.

continued on next page

WAIS over Z39.50 and Beyond *(continued)*

A search request primarily contains information on what database to search and the corresponding query, where the query may contain a natural language question, a Boolean expression, and a set of documents for use in *relevance feedback.* Relevance feedback is the ability to select a document, or portion of a document, and search for a set of documents similar to the selection. A search response is composed of a relevance-ranked list of *WAIS Citations,* where each citation contains summary information associated with the document. A WAIS Citation provides enough descriptive information on the document for a user to determine if a full retrieval is desirable. A retrieval request mainly contains a *document identifier* [6], which fully specifies the name and the location of the requested document. And finally, the corresponding retrieval response returns the desired document.

Design goals

As an aid to understanding the original WAIS implementation and its use of Z39.50-1988, the historical design goals of WAIS are presented in this section. Each design goal is accompanied by a brief description of the Z39.50 assumptions, the details of which are described in the next section.

- *Provide users with access to bibliographic and non-bibliographic information, including full-text and images:* Because Z39.50-1988 grew out of the bibliographic community, additional assumptions with the protocol were required to serve non-bibliographic information. They were also necessary to serve documents existing in multiple formats (e.g., RTF, *PostScript,* GIF, etc.).

- *Keep the search query simple and independent of changes in server functionality:* Most client implementations of Z39.50 parse the user's query into a Type-1 RPN (Reverse-Polish Notation) query where each term in the query is associated with a set of bibliographic attributes. For WAIS queries, a new Type-3 query type was assumed, which eliminated the need for the client to parse the query. The client was also liberated from the responsibility of knowing what bibliographic attributes were supported by the server.

- *Provide relevance feedback capability:* Since relevance feedback is specified in the search query, relevant documents were assumed to be part of the new Type-3 query type.

- *Permit the server to operate in a stateless manner:* In Z39.50, a search results in the creation and maintenance of a *Result Set* on the server, where subsequent retrieval requests are made with respect to this Result Set. In order for the server to operate statelessly in a WAIS system, an alternative approach was required to eliminate the need for maintaining Result Sets.

- *Provide the ability for a client to retrieve documents in pieces:* Because retrieval of a portion of a document could be done several ways with Z39.50-1988, assumptions were made to implement this functionality. Accessing a portion of a document was required for both retrieval and relevance feedback.

- *Run over TCP:* The Z39.50-1988 standard was designed to run in the application layer using the presentation services provided by the OSI (Open Systems Interconnection) Reference Model. Due to the popularity of TCP/IP and the Internet, WAIS was designed to run over TCP. Use of Z39.50 over TCP is described in [7].

The protocol

WAIS supports the *Init* and *Search Services* of Z39.50-1988, where each Service is made up of a request from the client followed by a response by the server. To meet the stated design goal of maintaining a stateless server, both the WAIS search and retrieval functions are implemented using the Z39.50 Search Service. The Z39.50 *Present Service* is not used for retrieval, since it requires that the server maintain state, or Result Sets, between operations.

Because the Z39.50 Search Service request contains a query and a corresponding query type, a WAIS search is distinguished from a WAIS retrieval by the query type. A WAIS search is implemented using the newly-defined Type-3 query, and a WAIS retrieval is implemented using a Z39.50 Type-1 query.

A WAIS search is initiated by the client with a Z39.50 Search Service Request APDU (*Application Protocol Data Unit*) using a Type-3 query. The query contains two main fields, the seed words and a list of document objects. The *seed words* contain the text typed by the user. A *document object* refers to a full document, or portion thereof, to be used in relevance feedback. Each document object contains a document identifier, type, chunk-code, and start and end locations. The *document identifier* and *type* specify the location and format, respectively, of the document. The *chunk-code* determines the unit of measure for the start and end locations. Examples of chunk-codes used include byte, line, paragraph, and full document. If the chunk code is a full document, the start and end locations are ignored.

A Search Service Response APDU returned by the server contains a relevance-ranked list of records, or WAIS Citations. A WAIS Citation refers to a document on the server. Each WAIS Citation contains the following fields:

- *Headline:* a set of words that convey the main idea of the document.
- *Rank:* the numerical score of the document based on its relevance to the query, normalized to a top score of 1000.
- *List of available formats:* e.g., text, *PostScript,* GIF, etc.
- *Doc-ID:* the document identifier for the document.
- *Length:* the length of the document in bytes.

The number of WAIS Citations returned is limited by the preferred message size negotiated during the Init Service.

A WAIS retrieval of a document is initiated by the client with a Search Service Request APDU using a Type-1 query. The query contains up to four terms: the Doc-ID, a document format, the start location, and the end location. The Doc-ID is obtained from the WAIS Citation sent to the client during a previous WAIS search, and the document format is selected by the user from the list of available formats supplied in the WAIS Citation. Because full-text and images are often larger in size than the receive buffer of the client, clients are designed to optionally retrieve documents in chunks, specifying the start and end positions of the chunk in the query. The Z39.50 Use and Relation bibliographic attributes taken from the Bib-1 Attribute Set are used to distinguish each term in the Type-1 query.

continued on next page

WAIS over Z39.50 and Beyond *(continued)*

The Use and Relation Attributes associated with the terms in the WAIS Type-1 query are specified as follows:

- Doc-ID — Use: system-control-number, Relation: equal
- Document format — Use: data-type, Relation: equal
- Start location — Use: paragraph, line, or byte, Relation: greater-than-or-equal
- End location – Use: paragraph, line, byte, Relation: less-than

The Use Attributes of data-type, paragraph, line, and byte are not part of the Z39.50 Bib-1 Attribute set, and are assigned the unique codes, "wt," "wp," "wl," and "wb," respectively. An example of a fully-specified retrieval query is:

```
query =
  ( ( Use = system-control-number, Relation = equal, term = <Doc-ID>
    AND
    ( Use = data-type, Relation = equal, term = postscript)
    AND
    ( use = byte, relation = greater-than-or-equal, term = 0 )
    AND
    ( use = byte, relation = less-than, term = 2000 ) )
```

A retrieval response is issued by the server with a Search Service Response APDU. In this case, a single record corresponding to the requested document, or portion thereof, is returned in the specified format.

The next generation

Since the first release of the WAIS system, the Z39.50 standard has been significantly enriched and its popularity has increased considerably as evidenced by the growing numbers of registered implementors, nationally and internationally. Representation within the Z39.50 group has expanded to not only include representatives from the librarian community, but also representatives from government agencies, educational institutions, and the commercial sector.

A number of new standards have emerged to meet the evolving needs of the networked information retrieval world. These new standards serve to complement Z39.50. For example, document identifiers can be specified by the URI (*Universal Resource Identifier*) standards [8,9]. Document formats could be given using MIME Content Types [10]. Languages and character sets also have corresponding standards [11,12]. Commercial systems require additional standards, such as security and authentication [13]. For wide-area acceptance by the masses, a complete information retrieval system should make use of a number of these open standards.

Activity is underway to develop the next generation WAIS systems based on these new standards. At the core of the next generation WAIS systems is the WAIS Profile of Z39.50-1992 which was approved by the OIW (*OSI Implementors Workshop*) SIGLA (*Special Interest Group in Library Applications*) in December 1993. It specifies full conformance with Z39.50-1992, and requires use of the versatile Z39.50 *Generic Record Syntax* [14]. Also built into the Profile is the flexibility to use many of the newly emerging standards. This next generation work is based on the same guiding principles as the original WAIS network publishing system: a wide-area networked-based information system for searching, browsing, and publishing, based on standards, easy to use, and flexible and growth oriented.

MARGARET ST. PIERRE is a consultant to WAIS Incorporated where her work focuses primarily on the development of the next generation WAIS systems. She is also a core team member of the Government Information Locator Service (GILS) project, and a participant in the ZIG, OIW SIGLA, and IETF. She is currently on a leave of absence from the Advanced Development Group at Thinking Machines Corporation where she was mainly responsible for simulation and verification of the Connection Machine Model CM-5. She received her B.S.E.E. from Santa Clara University, and S.M.E.E. and E.E. Degrees from MIT. She is also a member of IEEE and ACM. She can be reached as: saint@bluangel.com

References

[1] Kahle, B. & Medlar, A., "Information Service for Corporate Users: WAIS," *ConneXions,* Volume 5, No. 11, November 1991.

[2] National Information Standards Organization (NISO), "American National Standard Z39.50, Information Retrieval Service Definition and Protocol Specifications for Library Applications," Transaction Publishers; 1988.

[3] ANSI/NISO, "Z30.50-1992 (version 2) Information Retrieval Service and Protocol: American National Standard, Information Retrieval Application Service Definition and Protocol Specification for Open Systems Interconnection," 1992.

[4] "Z39.50 Version 3: Draft 8," October 1993, Maintenance Agency Reference: Z39.50MA-034.

[5] St. Pierre, M., Fullton, J., Gamiel, K., Goldman, J., Kahle, B., Kunze, J. A., Morris, H., & Schiettecatte, F., "WAIS over Z39.50-1988," Internet Draft, November 1993.

[6] Kahle, B., "Document Identifiers, or International Standard Book Numbers for the Electronic Age," September 1991. See `URL=<ftp://wais.com/pub/protocol/doc-ids.txt>`.

[7] Lynch, C., "Using the Z39.50 Information Retrieval Protocol in the Internet Environment," Internet Draft, November 1993.

[8] Berners-Lee, T., "Uniform Resource Locators," Internet Draft, July 1993.

[9] Weider, C., & Deutsch, P., "Uniform Resource Names," Internet Draft, October 1993.

[10] Borenstein, N. & Freed, N. "MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies," RFC 1521, Sept. 1993.

[11] Library of Congress, "USMARC Code List for Languages," 1989.

[12] Simonsen, K., "Character Mnemonics & Character Sets," RFC 1345, June 1992.

[13] Kohl, J. & Neuman, C., "The Kerberos Network Authentication Service (V5)," RFC 1510, September 1993.

[14] Special thanks to John A. Kunze of the University of California, Berkeley, for his pioneering efforts on the development of the Generic Record Syntax.

[15] Deutsch, P., "The Internet as Service Provider," *ConneXions,* Volume 7, No. 9, September 1993.

More information

For more information on WAIS, see:

`URL=<ftp://wais.com/pub/wais-inc-doc/txt/bibliography.txt>`.

Internet mailings lists on WAIS include: `wais-talk@wais.com`, and `wais-discussion@wais.com`. On the USENET, look in newsgroup `comp.infosystems.wais`. For freeware WAIS clients and servers, see `URL=<ftp://wais.com/pub/freeware/>`.

For additional information on freeware WAIS, contact Clearinghouse for Networked Information Discovery and Retrieval, 3021 Cornwallis Road, Research Triangle Park, NC, 27709, (919) 248-1499, or send questions to `freewais@cnidr.org`. For information on commercial WAIS products and services, contact WAIS Incorporated, 1040 Noel Drive, Menlo Park, CA 94025, (415) 327-WAIS, or send questions to `info@wais.com`.

Essay

Seven Steps to Cyberspace

by Jon Crowcroft, University College London

This article is about an imagined journey from the current Internet to Cyberspace. It is about whether we will make it, whether we should make it, and if so, what are the steps we could take to get there from here. First of all, where are we now? Last of all, where are we going? The Steps:

Step 1: The Internet

You are in a room. Next to you are a PC, a Mac and a UNIX workstation. You pick up a mouse, open a window... Is it a Mac window (NCSA *telnet*), a KA9Q *tn* window or an X window? Yes, and it is a window on the world, the world of... *Archie, WAIS, Gopher* etc. You type a string in the Search Term field and click on query. A pop-up says "73 items found, six exact matches, four fuzzy." You pick up three of the fuzzy matches... There is a message from Vint Cerf lying in the corner of your screen... It says, "light one of the matches and you will see by its glow a secret exit which leads up to the next level." You do this and find yourself in:

Step 2: The Matrix

You are in a large open office—there are fax machines, telephones, dealer terminals and screens, screens, screens everywhere. You pick up an old Panasonic remote control that is lying next to a 1960s British Trim-phone.

There is a fax from Vint Cerf lying in the corner of your screen... You pick it up and warm it slowly with the second of your matches. Despite care, it bursts into flames and shrivels to a pile of ashes on your keyboard—it covers enough letters for you to work out that if you hit the escape key 3 times with the ALT and Backspace characters in search of an author, you will find yourself in:

Step 3: Video World

You are in a video rental store. Near you are stacks and stacks of virtual cassettes, and stacks of video monitors—entire wall is full of them, just like in "The Man Who Fell To Earth."

On some of them are politicians arguing with each other about whether they should keep their jobs, and viewers arguing with politicians that they are going to lose their jobs when universal video franchise through the *Interactive Referendum Continuum* act comes in to force.

On one old Sony in the corner there is the image of Vint Cerf pointing at a shared whiteboard, which says "Click the middle button on your remote." You do so, and find yourself in:

Step 4: Games World

You are on a large flat plane... The ghosts of Super Mario and Sonic the Hedgehog, wander across the horizon... You are a frequent denizen of MUDDs, MOOs, from MU to Lambda and back again...

Vint Cerf runs up at the last second with an important message...it reads "Start a match company with you last fuzzy match, and you will make millions." With these, you can light your way to:

Step 5: Dollar World

You are in a vast vault...next to you are virtually zillions of Yen you have purloined from a zaibatsu... Vint Cerf is nowhere in sight...but his influence on the markets is such that you are eventually forced to sell up and ship out, to:

Step 6: Cyberversity

You are in a class of your own... School is out, out there in sense-net... Vint Cerf is chairing the plenary session...his advice is that you move on up, to:

Step 7: Cyberspace

You are in a room next to you are a... PC a Mac and a UNIX workstation—you are in a simulation of a late 80s Computer Science Lab There is a Virtual Cerf in the simulation. Yes, in Cyberspace, no one can see your screen...

Interword: Will we go there?

Yes.

Afterworld: Should we go there?

No.

JON CROWCROFT is a senior lecturer in the Department of Computer Science, University College London, where he is responsible for a number of European and US funded research projects in Multi-media Communications. A recent project just completed worked on protocol migration (Internet and OSI). He has been working in these areas for over 10 years. For the last 3 years he has also been consulting to the Bloomsbury Computing Consortium as a Senior Systems Analyst on the installation of a multi-campus distributed system. He graduated in Physics from Trinity College, Cambridge University in 1979, and gained his MSc in Computing in 1981, and PhD in 1993. He is a member of the ACM, the British Computer Society and the IEE. He will be general chair for the ACM SIGCOMM 94 symposium. He has never been to an IETF since they have become more fun to attend on the MBONE. He can be reached as: `J.Crowcroft@cs.ucl.ac.uk`

Coming in future editions of *ConneXions*

Make sure your subscription is current, we have lots of great articles in store for you for 1994:

- An article by Jack Kessler on the French *Minitel* system, a competitor to the Internet?
- A Special Issue on *The Next Generation IP* (IPng) with articles on all of the major IPng candidates, analysis of the address explosion problem and much more.
- Tracy LaQuey describing K–12 networking efforts, the Internet in every classroom?
- A description of the NSFNET Network Discovery Algorithms by Bill Norton.
- Mobile IP as seen by the Internet Engineering Task Force in general and Charlie Perkins in particular.
- An article on Advanced Distributed Network Management with Windows SNMP by Bob Natale.

So, check that expiration date (we've been letting you have a couple of "grace" issues) and contact us to renew:

ConneXions—The Interoperability Report
303 Vintage Park Drive,
Suite 201
Foster City, CA 94404–1138
USA
Phone: +1 415-578-6900 or 1-800-INTEROP (Toll-free in the USA)
Fax: +1 415-525-0194
E-mail: `connexions@interop.com`

Book Reviews

Open Systems Networking: TCP/IP and OSI, by David M. Piscitello and A. Lyman Chapin, ISBN 0-201-56334-7, Addison-Wesley, 1993.

Luminaries

With the recent explosion of interest in the Internet has come an extraordinary profusion of books about its technology. One of the most recent offerings is by two luminaries, whose OSI credentials date back to its start. Messrs. Piscitello and Chapin provided seminal contributions to the OSI effort, most notably in its architecture and lower layer protocols. More recently, they have become my colleagues on the *Internet Engineering Steering Group* which oversees Internet standards development. Hence, the authors have a depth and breadth of experience that should bode well for analysis and perspective. By and large, that is the major strength of this book.

Organization

The book is extremely ambitious. It provides descriptions of architecture, protocol details, comparisons between OSI and TCP, and description and commentary about standards making, all in an extremely readable style. Small sections are marked "AHA" to warn the reader of personal commentary, but the main text goes beyond simple exposition, too. The first seventy pages are devoted to discussion of architecture and concepts, most of which were quite well done though I have never thought of *Abstract Syntax Notation One* (ASN.1) as a "language." In any event, anyone who devotes a distinct chapter to discussing naming and addressing gets extra points from me.

As expected, the content covering the lower layers of OSI is excellent. The book is rich in detail about the specification effort and has an extensive chapter on routing. They also provide constant commentary about the choices and difficulties in creating the standards. Often this includes very pointed criticisms of the choices made; such candor is rare and welcome. Discussion of OSI's upper layers is notably weaker in assorted technical details, but is well organized, thorough, and makes for a good introduction.

Soup test

Some people assess a French restaurant based on its rendition of onion soup which is a relatively simple dish, ruined remarkably often. I like to do a quick check of a TCP/IP book by reading its description of the *TCP Urgent Data* mechanism which is highly idiosyncratic. The authors do better than average and get credit for providing quite a bit of detail; unfortunately key portions are wrong. One problem also occurs in a few other places in the book: a slight tendency to mix discussion of implementation with discussion of protocol without distinguishing them. Overall, the sundry difficulties with details of the TCP/IP stack and its standards process mean that this book should not be used as a first-encounter with that technology, but it is fine for follow-on reading.

An OSI book

Given that TCP and OSI remain major players in the life of data networking professionals, there is great benefit in being able to read about them in a combined text, particularly when the discussion includes the authors' opinions about the two stacks. It is curious that so much of the commentary freely and frequently admits to the various failings of the OSI technology, but repeatedly and inevitably concludes that its future remains bright. I'm tempted to suggest that the authors are looking at things through Rose colored glasses, but Marshall might not agree. In truth, this is an OSI book with a great deal of TCP/IP content.

Two-way discussion

Some books should be read as part of a one-way communication process, in which the author conducts a monologue, filling the reader's brain with new facts and figures. Other books work best as a two-way discussion between the author and the reader, the author triggering reactions of surprise, agreement, disagreement, and further thought. *Open Systems Networking* is such a book. If you count yourself as a professional in data networking, you need to develop a broad perspective on both architecture and protocol details. This book will help you.

—Dave Crocker, Silicon Graphics

Connecting to the Internet, a Buyer's Guide, by Susan Estrada, O'Reilly & Associates, Inc., August 1993, ISBN 1-56592-061-9.

Background

The Internet is a western cultural phenomenon. It has achieved this status in part because it does what it claims to, connecting many diverse computing environments into a common set of communications services. Lots of folks are learning about the Internet since some enterprising people have started distribution of lots of information in this environment and an equally motivated group has gone off and done a great PR job, telling all sorts of people what a great thing the net really is. The beginnings of mass marketing of the Internet is on us. There are a large number of books, guides, seminars and the like, telling people what the net is and what kind of potential is there. This activity has filled part of the void, to tell people what to do once they get on. This book fills the other part of the void, how to find a provider and how to select one based on reasonable criteria.

Organization

Susan takes a different tack than most other books on the Internet. She starts out with the basics of information transfer and then moves on into a non-technical discussion on the end-user perception of performance. Once these concepts are given, she moves into a structured method of requirements analysis that enable the reader to assess their networking needs. The user should now have the tools needed to jump into the next section of the book, provider selection. She then closes out with yet another list of service providers.

From where I sit

This book provides two areas of focus that heretofore have been missing from the Internet Lexicon. For MIS managers, this slim volume provides a place to "sanity check" the hype and hysteria that seem to accompany demands for Internet access. It will give rational guidelines when meeting with users so that appropriate access can be sought out. If this was the only component that the book had, it would be worth the price. Everyone who has to acquire services from a provider should get a copy for that alone. However, the book has another facet that is of use to Internet Service Providers. For the first time, there is an objective checklist that customers can use to rate providers. Astute providers will pick up a copy so they can rate themselves, target market segments, and find niches that are not covered in the book. This is the fourth "Internet" book to land a place on my bookshelf. . . unless someone else has borrowed it. . . again.

—Bill Manning, Rice University

[Ed.: See the next two pages for a list of Internet books——>]

Books about the Internet

Introduction

The last couple of years has seen the publication of many books about the Internet, and new ones seem to appear almost monthly. Here is a list of the books we are aware of as of late January 1994, but this list is certain to be out-of-date by the time it reaches you. The books listed are Internet-specific in the sense that explain what the Internet is, how to get connected and how to use it. There is of course a whole set of TCP/IP protocol- and application-specific books as well, but these are not included in this list. Special thanks to the folks at Computer Literacy Bookshops for their help in compiling this summary. We also recommend that you join The Internet Society for regular updates on Internet developments through their *Internet Society News.*

- *The Whole Internet User's Guide and Catalog* by Ed Krol, O'Reilly & Associates, ISBN 1-56592-025-2. [Reviewed in *ConneXions,* December 1992].
- *The Internet Companion: A Beginner's Guide to Global Networking* by Tracy LaQuey, Addison-Wesley, ISBN 0-201-62224-6. [Reviewed in *ConneXions,* December 1992].
- *Internet: Getting Started* by April Marine et al, Prentice-Hall, ISBN 0-13-289596-X.
- *Exploring The Internet: A Technical Travelogue* by Carl Malamud, Prentice-Hall, ISBN 0-13-296898-3. [Reviewed in *ConneXions,* October 1992].
- *Zen and the Art of the Internet: A Beginner's Guide* by Brendan P. Kehoe, Prentice-Hall, ISBN 0-13-010778-6. [Reviewed in *ConneXions,* August 1992].
- *Crossing the Internet Threshold: an Instructional Handbook* by Roy Tennant, John Ober, Anne Lipow, Library Solutions Press, ISBN 1-882208-01-3. [Reviewed in *ConneXions,* November 1992].
- *The Internet Navigator* by Paul Gilster, John Wiley & Sons, ISBN 0-471-59782-1.
- *Navigating the Internet* by Mark Gibbs and Richard Smith, Sams, ISBN 0-675-30362-0.
- *The Internet: Complete Reference* by Harley Hahn and Rick Stout, Osborne/McGraw-Hill, ISBN 0-07-881980-6.
- *The Internet Guide for New Users* by Daniel Dern, MacGraw-Hill, ISBN 0-07-016510-6.
- *The Internet Connection: System Connectivity and Configuration* by John Quarterman, and Smoot Carl-Mitchell, Addison-Wesley, ISBN 0-201-54237-4.
- *Connecting to the Internet, a Buyer's Guide,* by Susan Estrada, O'Reilly & Associates, ISBN 1-56592-061-9. [Reviewed in *ConneXions,* February 1994].
- *Internet Primer for Information Professionals: A Basic Guide to Networking Technology* by Elizabeth Lane and Craig Summerhill, Meckler, ISBN 0-88736-831-X.

- *The Internet Passport: NorthWestNet's Guide to Our World Online* by Jonathan Kochmer, NorthWestNet, ISBN 0-9635281-0-6.
- *The Internet Roadmap* by Bennett Falk, Sybex, ISBN 0-7821-1365-6.
- *Riding the Internet Highway* by Sharon Fisher, New Riders Publishers, ISBN 1-562-05192-X.
- *The Instant Internet Guide* by Brent Heslop & David Angell, Addison-Wesley, ISBN 0-201-62707-8.
- *Internet Basics* by Steve Lambert and Walt Howe, Random House, ISBN 0679-75023-1.
- *The Internet for Dummies* by John Levin and Carol Baroud, IDG, ISBN 1-56884-024-1.
- *Welcome to Internet: From Mystery to Mastery* by Tom Badgett and Lorey Sandler, MIS Press, ISBN 1-55828-308-0.
- *Internet for Everyone* by Richard Wiggins, McGraw-Hill, ISBN 007-067019-6.
- *Bridging the Internet Gap: How to make the world your on-line oyster* by James Potter, Bridge Learning Systems, ISBN 0-9632069-8-2.
- *Internet World's* "On Internet 94" by Tony Abbott (Editor), Meclermedia, ISBN 0-88736-929-4.
- *The Internet System Handbook,* by Dan Lynch and Marshall Rose (Editors), Addison-Wesley, ISBN 0-201-56741-5. [Reviewed in *ConneXions,* March 1993].
- *Mac Internet Tour Guide* by Michael Fraase, Vantana Press, ISBN 1-56604-062-0.
- *PC Internet Tour Guide* by Michael Fraase, Vantana Press, ISBN 1-56604-084-1.
- *Windows Internet Tour Guide* by Michael Fraase, Vantana Press, ISBN 1-56604-146-5.
- *A DOS User's Guide* to the Internet by James Gardner, Prentice-Hall, ISBN 0-13-106873-3.
- *The Internet Directory* by Eric Braun, Fawcett Columbine, ISBN 0-449-90898-4.
- *Directory of Directories on the Internet* by Gregory Newby, Meckler, ISBN 0-88736-768-2.
- *Doing Business on The Internet* by Mary Cronin, Van Nostrand Reinhold, ISBN 0-442-01770-7.
- *The Canadian Internet Handbook* by Rick Broadhead & Jim Carroll, Prentice-Hall Canada, ISBN 0-13-304395-9.

Network Management Experts Proclaim SNMP Test Summit A Success

SNMP Compatibility Tests To Be Made Freely Available to Promote Self-Policing of SNMP Implementations

San Jose, California January 17, 1994—Engineers participating in the first *SNMP Test Summit* held last week at the Center for Software Development were very enthusiastic about the results of the interoperability Summit. The SNMP Test Summit was created to help vendors exercise SNMP engines, agents and managers incorporated in software and hardware products for protocol correctness. Among the participants were some of the leading experts in network management, including engineers from Cabletron Systems Inc., Cisco Systems Inc., Eicon Technology, Empirical Tools & Technologies, Epilogue Technology Corp., Fujitsu OSSI, IBM and IBM Research, Network General Corp., PEER Networks, SNMP Research, SynOptics Communications, TGV, Inc., and Wellfleet Communications. The SNMP testing Summit was organized by InterWorking Labs, "The Protocol Police of the Information Highway."

Objectives

"When we decided to hold the SNMP Test Summit, our objective was to create a high-quality, open, unbiased environment where we could police ourselves with regard to standards compliance," said Chris Wellens, President of InterWorking Labs. "The Summit provides an ideal forum where the experts can compare notes and exchange ideas. There's no fear of failure here, or of tallying winners and losers. Instead, we offer an environment where all can work together to solve the real problems facing us when we implement new networking standards. To help vendors continue their self-policing efforts, we have decided to make the test suites freely available."

"The summit provided a valuable opportunity for vendors to test SNMP implementations in an unbiased, neutral testing environment," said Mark Truhlar, Cabletron's Director of Product Development, Network Management. "Our customers are truly the ones who benefit from events such as this one as vendors work together to ensure product interoperability."

Real winners

"The real winners are SNMP customers," added Karl Auerbach, President of Empirical Tools & Technologies. "We all found some kind of bug in our SNMP implementations, but with a common set of compatibility tests, it will be that much easier to create SNMP products that make customer networks truly interoperable." "I appreciate the existence of a community forum where companies can ferret out common bugs and later have the testing suite available for regression purposes," said Robert Snyder, a member of the development team at Cisco Systems.

Participants had universal praise for the quality of the SNMP test suites. "We found a lot of problems that our internal testing never uncovered," noted one developer. "These tests were well-designed, systematically covering a wide range of boundary conditions," added an engineer from PEER Networks.

"This SNMP Test Suite provides a model for protocol testing that would be valuable for all protocols in the Internet environment," said David Bridgham, a member of the SNMP development group at Epilogue Technology.

"If you're serious about using SNMP to solve your network problems, you must have confidence in the answers your agent gives you," said Marshall T. Rose, Principal of Dover Beach Consulting, Inc., and Area Director for Network Management in the Internet. "By getting together and performing these tests, all of us are able to discuss and agree about how to handle certain problems."

Benefits

Participants also agreed that one of the key benefits to emerge from the SNMP Test Summit was that it gave competitors an opportunity to discuss the challenges of implementing new *Simple Network Management Protocol* (SNMP) technologies. For example, engineers from SNMP Research, PEER Networks, and Epilogue discussed the challenges of implementing SNMPv2 over lunch, and experts from Cabletron and SynOptics compared notes on the appropriate behavior for SNMP managers dealing with pathological agent failures. Many of the vendors also exchanged ideas on the future direction of SNMP, such as developing a common *Application Programming Interface* (API), for extensible SNMP agents, using the administrative framework for SNMPv2, and possible action on creating a MIB API. "We found the first round of tests and the opportunity to exchange ideas with other vendors very useful and look forward to the next round of tests focusing on sub-agents and the network managers' perspective," said Vartan Narikian, Technical Leader at Eicon Technology. "I'm looking forward to the next Summit where we can extend the manager side of the tests to use expert systems principles in evaluating appropriate behavior," noted an engineer from SNMP Research.

More information

The next SNMP Test Summit is scheduled to take place the week of June 27, 1994. A revised implementation of the tests is planned to be made available in May. This will allow participants to pretest their SNMP managers and agents before bringing products to the SNMP Test Summit for interoperability testing. For more information about the test specifications, contact InterWorking Labs. InterWorking Labs is an independent testing service that provides Software Quality Assurance (SQA) and System Integration Engineering (SIE) services to computer hardware and software manufacturers. The company has offices at 479 Yosemite Avenue, Mountain View, CA 94041-2154, USA; Telephone: (415) 969-4544; Fax: (415) 965-3459.

A set of SNMP Test Event Specifications is available upon request.

Write to *ConneXions*!

Have a question about your subscription? Are you moving, and need to give us your new address? Suggestions for topics? Want to write an article? A letter to the Editor? Have a question for an author? Need a *ConneXions* binder for your back issues or perhaps a *ConneXions* sweatshirt? Want to order some back issues? (there are now over 80 to choose from; ask for our free 1987–1992 index booklet and the 1993 index sheet). We want to hear from you. Send your questions, comments or suggestions to:

ConneXions—The Interoperability Report
303 Vintage Park Drive
Suite 201
Foster City, CA 94404–1138
USA
Phone: +1 415-578-6900 or 1-800-INTEROP (Toll-free in the USA)
Fax: +1 415-525-0194
E-mail: `connexions@interop.com`

Call for Papers

The *First International Workshop on Community Networking: Integrated Multimedia Services to The Home* will be held July 13–14, 1994 at the Westin Hotel, in San Francisco, California, USA. The workshop is sponsored by the IEEE Communications Society in collaboration with ACM SIGCOMM, the Internet Society, and Smart Valley.

Background

Community networking concerns the network infrastructures that will bring integrated multimedia services to home users. Community networking differs in many ways from enterprise networking in its services, technologies, and economics. In contrast to enterprise networking applications, community networking services will not necessarily be work oriented and will range from entertainment to shopping to information services. At present, community networking technology is driven by the requirements of video-on-demand, most notably high bandwidth (compared to narrowband), bandwidth asymmetry, and the delay-jitter constraints imposed by today's limited-storage TV set-top devices. As various other services develop, community networking will evolve to include integrated multimedia communication and user-to-user applications. Community networking must also provide access to resources located outside the community, in an increasingly global repository of information of every conceivable type. Since very little has been published to date on the topic of community networking, this workshop will give researchers and professionals the chance to share their views and advance the state of the art in this field.

Relevant areas

Contributions are encouraged in the four areas listed below with relevant topics:

- *Applications and Requirements:* types of applications; coding; set-top operating systems; QoS networking requirements (symmetric/asymmetric bandwidth, delay, and losses); security and privacy; service models; user interface and navigation facilities.
- *Local Distribution Technology:* topology; fiber/cable/UTP/wireless; modulation, bandwidth allocation; MAC (reverse channel); role of ATM; dependencies on equipment/network in the home (e.g., TV set-top).
- *Addressing, Signalling, and Upper-Layer Protocols:* local vs. global addressing; the service provider view vs. the common carrier view; the video-dialtone gateway; role of B-ISDN protocols; network- and transport-layer protocols; network management; APIs.
- *Internetworking and Architecture: the gateway:* accessing other networks (data, telephone); server placement and network optimization; the regional distribution centers; testbeds; network traffic models; network cost structure and its implications on service pricing; medium- and long-term network evolution; the impact of regulatory constraints.

Instructions for submitting abstracts

Please send via electronic mail a short abstract (up to 700 words in ASCII or *PostScript*) describing a position statement in one of the areas above to `cn-workshop@opera.hpl.hp.com.` Note: submissions longer than the limit above will not be reviewed. Only if electronic submission is impossible, a hardcopy version may be sent to:

Riccardo Gusella
Hewlett-Packard Laboratories
1501 Page Mill Rd., MS 1U-17
Palo Alto, CA 94304, USA

Participation in the workshop will be by invitation only based on the Program Committee's review of position statements. Some of the authors will be asked to submit extended abstracts and to present their positions during the workshop. Workshop size limitation may preclude attendance of all authors of multi-author abstracts.

Important dates

Deadline for submitting abstracts	April 15, 1994
Acceptance notification	May 12, 1994
Extended abstract due (limited to 2000 words)	June 16, 1994

Call for Papers

The *Third International Conference on Computer Communications and Networks* (ICCCN '94) will be held September 11–14, 1994, in San Francisco, California, USA. The conference is sponsored by USC Communication Sciences Institute in cooperation with the IEEE Communication Society.

Topics

ICCCN '94 will cover all aspects of computer communications and networks. Of particular interest are recent advances in high-speed distributed multimedia applications, and the design and performance evaluation of system and network architectures that support these applications. Authors are invited to submit papers and tutorial proposals. Topics of interest include, but are not limited to:

- ATM Network Design
- ATM Traffic Control
- ATM LANs
- ATM Testbeds
- Quality of Service Issues
- Wireless Networks
- Multicast Protocols
- Optical Networks
- Protocols for High Speed Networks
- Distributed Multimedia applications
- Multimedia Man–Machine Interface
- Video-on-Demand Systems
- Video Coding
- Network Management
- Performance Modeling/Analysis
- LAN/WAN Internetworking

Instructions to authors

All submissions must be accompanied by a cover letter containing a list of all authors, their affiliations, phone/fax numbers, and e-mail addresses. Papers should be at most 20 double-spaced pages and must include an abstract of 100–150 words with up to ten keywords. Tutorial proposals should contain a one-page description and a detailed outline. All submissions will be refereed and judged with respect to quality and relevance. Submit six copies of each paper to the Program Chair and tutorial proposal to the Program Vice Chair.

Professor J. W. Wong, *Program Chair*
Department of Computer Science, University of Waterloo
Waterloo, Ontario, CANADA N2L 3G1
Phone +1 (519) 888-4431; Fax (519) 885-1208
E-mail: `jwwong@math.uwaterloo.ca`

Professor E. K. Park, *Program Vice Chair*
Computer Science Department, United States Naval Academy
Annapolis, MD 21042, USA
Phone: +1 (410) 267-3037; Fax (410) 267-2686
E-mail: `eun@usna.navy.mil`

Important dates

Deadline for paper and/or tutorial submissions:	March 4, 1994
Notification of acceptance:	May 10, 1994
Camera ready papers due:	June 10, 1994

Call for Papers

Multimedia Transport and Teleservices will be held November 14–15, 1994 in Vienna, Austria. The conference is organised by the CEC COST 237 Multimedia Telecommunications Services Project and hosted by Alcatel Austria AG.

Background

Although many distributed multimedia applications now exist as pilot projects on local networks, these prototypes have yet to be translated into realistic applications running over large scale heterogeneous high-speed networks. To help bring about this important transition, a number of initiatives such as the COST 237 Multimedia Telecommunications Services project in Europe and the Multimedia Communications Forum in the US have recently been established. These groups identify a lack of generic system support as the primary technological factor holding back the deployment of realistic, large scale, distributed multimedia applications. There are two basic technologies required to make feasible such support: an appropriate transport service for communications needs, and a suitable set of generic multimedia teleservices to provide a framework for application development.

It is now accepted that significant enhancements to existing transport services are needed to adequately support large scale distributed multimedia applications. In particular, the transport service must be extended to support quality of service configurability and multicast/multipeer connectivity, and must be supported by a variety of high-speed network architectures. The area of multimedia teleservices is equally crucial. Generic high level services, such as multimedia enhanced e-mail, conferencing frameworks and shared application frameworks, are necessary to ease the evolution from present day pilot applications to commercial, interoperable, products. The conference will address both of these technological areas with particular attention paid to the integration of the two. The emphasis of the conference will be on service and architectural aspects of distributed multimedia application support from the transport layer upwards.

Topics

Important topics for the conference include (but are not limited to):

- Teleservices architecture
- Interfacing teleservices to applications
- Multimedia enhanced e-mail
- Wide-area multimedia collaboration tools
- Media synchronisation services
- Multimedia enhanced transport services
- Transport layer quality of service support
- Multipeer transport services
- Impact on OSI and TCP/IP
- Multimedia in TINA, ODP, DCE etc.

Instructions for submitting papers

Papers should be less than 20 pages long, single spaced, and be submitted in *PostScript* format. Authors should submit their papers by electronic mail to `cost237-conf@comp.lancs.ac.uk`. If electronic submission is impossible, papers (6 copies) may be sent to: Conference Secretary, Room B4, Computing Dept., Lancaster University, Lancaster LA1 4YR, UK. (Fax: +44 524 38 1707; Phone +44 524 59 3798).

Important dates

Papers due:	April 10, 1994
Acceptance notification:	July 10, 1994
Final paper due:	September 10, 1994

Call for Papers

Introduction

The international research journal *Computer Communications* publishes research, reviews, tutorials, case studies and application notes on a spectrum of data communications topics ranging from broadband through ATM and distributed systems to security, network management, multimedia, operating systems and protocols. As part of the journal's policy to publish key research in the field, our publishing programme includes an extensive series of special subject issues.

Special Issues

The journal is currently soliciting original R&D contributions for publication in its special issue programme in 1994/5. Special issues scheduled for publication during the next 18 months are:

- *June 1994:* Network security (Guest Editor, Prof. Sead Muftic, Stockholm University, Sweden)
- *September 1994:* Distributed systems (Guest Editor, Dr Ahmed Tantawy, IBM TJ Watson Research Center, NY, USA)
- *December 1994:* Multimedia storage and databases (Guest Editor, Prof. Tom Little, Boston University, MA, USA)*
- *January 1995:* Electronic document delivery (Guest Editor, Dr Lilian Ruston, Bellcore, MA, USA)*
- *March 1995:* Mobile computing (Guest Editor, Prof. R Badrinath, Rutgers University, USA)*
- *July 1995:* System support for mobile computing (Guest Editor, Prof. Kevin Jeffay, University of North Carolina at Chapel Hill, USA)*

Those issues marked * above are seeking high quality R&D contributions on the subject areas in question. All contributions will be subject to the journal's usual refereeing process. Full calls for papers can be obtained from the journal's General Editor or from the Guest Editors.

More information

Further information on *Computer Communications* special issues, and on our regular publishing programme, can be obtained from the General Editor:

Jeremy B Thompson
General Editor
Computer Communications
PO Box 31
Market Harborough
Leics LE16 9RQ
United Kingdom
Tel: +44 858 86382
Fax: +44 858 86635
E-mail: `100113.2636@compuserve.com`

CONNEXIONS
303 Vintage Park Drive
Suite 201
Foster City, CA 94404-1138
Phone: 415-578-6900
FAX: 415-525-0194

FIRST CLASS MAIL
U.S. POSTAGE
PAID
SAN JOSE, CA
PERMIT NO. 1

ADDRESS CORRECTION
REQUESTED